Amendments to the Specification

Please replace the paragraph on page 10, lines 11-23 with the following amended paragraph:

The packet initially enters linecard control element 130 at receive FIFO 215. This FIFO buffer is employed to absorb bursts of small packets which arrive while the pipeline isn't moving fast enough to keep up. Additionally, receive FIFO 215 allows control element 130 to process packets from multiple input interfaces 111. Referring to Figure 3, incoming packets 113 are then separated into a header portion and a tail portion by byte counter 310, a part of receive FIFO 215. Receive FIFO 215 comprises two logically distinct FIFOs. Header portions, here simply defined as the first *n* bytes of the received packet, are placed in header FIFO 320. The balance of the packet, i.e., bytes *n*+1 through the end of the packet, are placed in tail FIFO 330. Here the term "header portion" refers only to the first *n* bytes of the packet; it is not synonymous with the packet header defined as part of a packet protocol. The length of the header portion *n* is selected so that it includes the entire protocol header for the types of packets expected ~~on~~ in the device.

Please replace the paragraph on page 13, lines 24-28 with the following amended paragraph:

Outbound (transmit) packets are serviced from these output queues by packet transmit stage ~~1226~~ 1526 (and sent to the various interfaces) via a bandwidth-sharing algorithm such as MDRR. In the process, packets have their output encapsulations added as the packet is read from transmit packet buffer 285. The encapsulation is determined by a field of the BHDR, which was set in pipelined switch 220.

Please replace the paragraph on page 14, lines 1-4 with the following paragraph:

Transmit FIFO 1430 (referring to Figure 2) is between the packet transmit stage ~~1226~~ 1526 and network physical interface 210 to absorb bursts and keep the utilization of the media links at 100%. The packet 114 thus leaves control element 130 through network physical interface 210, eventually leaving linecard 110 for network 1.

Please replace the paragraph on page 16, line 26 through page 17, line 4 with the following paragraph:

Pipelined switch 220 also maintains, in one embodiment of the present invention, two sets of registers for inter-stage communications, the packet information registers (PIRs) and the packet control registers (PCRs). There are five PIRs (530, 640, 780, 880, and 940) and five PCRs (540, 650, 790, 890, and 950), one each for each pipeline stage, shown in Figs. 5 through ~~10~~ 9. The PIRs propagate information about the packet to be used for normal fast path processing. The PCRs, by contrast, only propagate control and exception information. Normally, the PCRs are empty (all zeros). Flags in the PCRs are only set if an exception or error condition is encountered in a stage; the flags are then propagated in the PCRs down the pipeline.

Please replace the paragraph on page 17, lines 15-27 with the following paragraph:

As fetch stage 410 receives the packet header it ~~perform~~ performs the following operations. First, the FS writes the packet header into the PHB starting at the PHB offset address contained in a unique pointer assigned by the pipeline control. Next, the fetch stage calculates the IP header checksum in IP checksum validator 510. In one embodiment of the present invention, the FS assumes that the packet is IP with 4 bytes of MAC encapsulation preceding it and calculates the IP header checksum of the incoming packet accordingly. (One of ordinary skill in the art of course appreciates that such an assumption is not required but is only made because it covers the majority of typical cases. Accordingly, the present invention is not so limited.) Based on the checksum calculation, FS 410 sets an indication in the PCR 540 if the checksum is correct or not, which is used by PreP stage 420. If the packet is not IP, this indication is ignored. If the packet is IP and the checksum is incorrect, the packet is discarded as described further below.

Please replace the paragraph on page 34, lines 21-28 with the following paragraph:

Each octet in the IPv4 address represent a different layer in the tree. See Figure 16. The first octet belongs to the first layer 1610 which has 256 entries, one for each value of the 8 bits. Each entry ~~contain~~ contains a pointer that either points to the starting address of the second layer 1620 (i.e., L=0, the entry represents a node) or points

to a leaf address (L=1). The second layer for each of the first layer entries has also 256 entries and each entry can also either point to a leaf or is a node and thus points to the third layer 1630. Again, each third layer node has 256 entries. Pointers in the third layer can either point to a leaf or point to a node. All entries in the fourth layer point to a leaf.